

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-232856

(43) 公開日 平成10年(1998) 9月2日

(51) Int.Cl.⁵
G 0 6 F 15/16

識別記号
3 9 0

F I
G 0 6 F 15/16

3 9 0 Z

審査請求 未請求 請求項の数 3 O L (全 12 頁)

(21) 出願番号 特願平9-33710

(22) 出願日 平成9年(1997) 2月18日

(71) 出願人 000005267

ブラザー工業株式会社

愛知県名古屋市瑞穂区苗代町15番1号

(72) 発明者 村松 喜世治

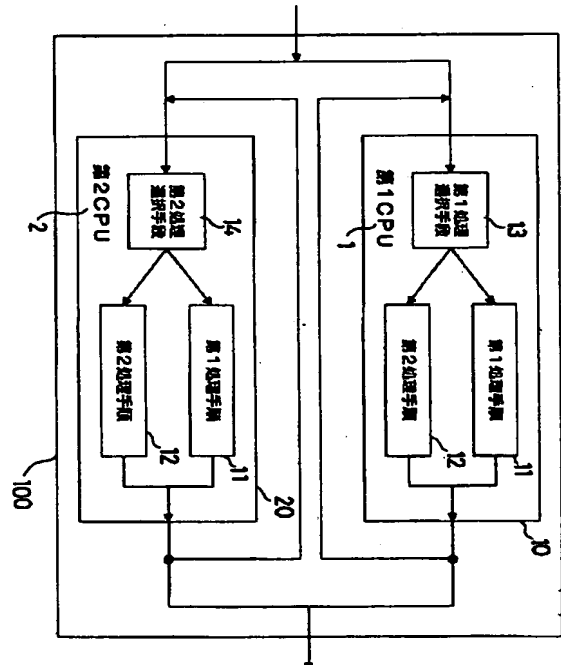
名古屋市瑞穂区苗代町15番1号ブラザー工業株式会社内

(54) 【発明の名称】 並列処理方式

(57) 【要約】

【課題】 ストール時間を生じること無く、柔軟なスケジューリングを可能とすることによって高速に動作することが可能な並列処理方式を提供すること。

【解決手段】 並列処理装置100は、第1複合処理ユニット10と、第2複合処理ユニット20とを備える。第1複合処理ユニット10及び第2複合処理ユニット20は、予め処理手順が定められた単体処理ユニットとしての第1処理手順11及び第2処理手順12と、予め第1処理手順11または第2処理手順のいずれを優先的に選択するかを定義した単体処理ユニット選択順序に従って、前記第1処理手順11または第2処理手順12の何れか一方を選択する第1処理選択手段13とから構成される。



1

【特許請求の範囲】

【請求項 1】 処理手順の内容が同一あるいは異なる複数の単体処理ユニットと、その複数の単体処理ユニットの中から実行すべき単体処理ユニットを選択する単体処理ユニット選択手段とから構成される、複数の複合処理ユニットと、前記複数の前記複合処理ユニットを並列的に実行する並列処理制御手段と、を備えることを特徴とする並列処理方式。

【請求項 2】 前記単体処理ユニット選択手段は、予め定義された優先順位である単体処理ユニット選択順序に従って、前記複数の単体処理ユニットの中から実行すべき単体処理ユニットを選択することを特徴とする請求項 1 に記載の並列処理方式。

【請求項 3】 前記単体処理ユニット選択手段は、前記単体処理ユニット選択順序に従って選択された前記単体処理ユニットが実行可能であるか否かを判定する単体処理ユニット実行可能性判定手段と、実行可能な前記単体処理ユニットが選択されるまで単体処理ユニットの選択動作を行うように前記単体処理ユニット選択手段を制御する単体処理ユニット選択制御手段とを備えることを特徴とする請求項 1 または 2 に記載の並列処理方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数のプロセッサにより並列処理を実行する並列処理方式に関するものである。

【0002】

【従来の技術】 従来、並列処理によって実行時間を短縮しようとする方法は多数提案されてきた。

【0003】 例えば、ベクトル演算処理に於ける並列処理がある。

【0004】 ベクトル演算処理に於いて、例えば、下記のような制御ループがあった場合を考える。

【0005】

【数 1】

```
for( i=1 ; i<n ; i++ ) {  
    a[i] = a[i] + a[i-1];  
    b[i] = b[i] + b[i-1];  
}
```

【0006】 この場合に、これを並列処理が可能のように、下記のように分割する。

【0007】

【数 2】

2

```
for( i=1 ; i<n ; i++ ) {  
    a[i] = a[i] + a[i-1];  
}  
  
for( i=1 ; i<n ; i++ ) {  
    b[i] = b[i] + b[i-1];  
}
```

【0008】 分割された前記の各制御ループに対して別々のプロセッサを割り当て、並列処理を実行することによって、実行時間の短縮を実現していた。

【0009】 前記のベクトル演算処理に限らず、並列処理を行う場合は、並列処理を行う対象となる処理を、互いの処理結果に依存しない複数の処理ユニットに分割し、分割後の各処理ユニットを別々のプロセッサで並列処理することで、実行時間の短縮を実現する方法が提示されていた。

【0010】 しかしながら、前記の並列処理方式に於いては、分割された処理ユニットが互いの処理結果に依存しないことが前提となっているため、前の処理の結果を受けて行う処理に於いては、並列処理の効果が得られない問題があった。

【0011】 この問題に対し、並列処理に於いてスケジューリングを行うことによって、前の処理の結果を受けて行う処理についても並列処理の効果が得られる構成の並列処理方式が従来より提案されている。

【0012】 図 11 は、その一例として、2 つの処理 A、B をスケジューリングし、並列処理を実行した場合のタイムチャートを示した図である。

【0013】 ここで、処理 B (i) (i=1、2、3…) は、処理 A (i) (i=1、2、3…) の結果に依存した処理であり、処理 A (i) は処理 B (i) の結果に依存しない処理であるとする (以下同様)。

【0014】 この場合、処理 B (1) は処理 A (1) より先に開始することができないが、処理 A (2) は処理 B (1) の処理終了を待つ必要はないため、処理 A (1) の処理終了直後に処理を開始することが可能である。このように、処理 B の開始時間について矛盾が生じないようにスケジューリングすることにより、並列処理による実行時間の短縮が可能となる。

【0015】 図 11 (a) は、処理 A の実行時間が処理 B の実行時間よりも長い場合のタイムチャートであり、図 11 (b) は、処理 B の実行時間が処理 A の実行時間よりも長い場合のタイムチャートを示している。いずれの場合も、並列処理を行わない場合よりも全体の実行時間は明らかに短縮される。

【0016】

【発明が解決しようとする課題】 しかしながら、上述の例のようにスケジューリングを行って並列処理を実行す

3

る場合、図11(a)に示す如く、プロセッサ2に、次の処理が開始可能となる条件が満たされるまで何もできずに停止する(ストールする)時間(以下、ストール時間と称する)が発生してしまう問題があった。このストール及びストール時間は、プロセッサ資源の一部を遊ばせ、プロセッサの有する処理能力を十分に発揮させることができない問題を生じさせていた。

【0017】尚、図11の処理A、Bが、オペレーティングシステム(以下、OSと記述する)の上で動作するアプリケーションプログラムである場合、図11(a)のストール時間中にOSの処理を挿入することも可能ではある。この場合、プロセッサ2のストール時間は減少し、2個のプロセッサは休みなく動作していることにはなるが、アプリケーションプログラム自体の実行時間が短縮されるものではない。

【0018】また、図11(b)の場合は、プロセッサ1、プロセッサ2が休みなく動作しており、これ以上のストール時間は発生しないように見える。しかし、図示のように、処理Bの実行時間が処理Aの実行時間より長い場合には、先に行われた処理A(i)の結果を記憶領域に格納し、その記憶領域に格納された処理A(i)の結果を読み出して処理B(i)を実行することになるため、処理Bの処理が遅いと、処理Aの実行結果を格納する記憶領域がいっぱいになり、処理Bの実行が処理Aに追いつくまで(記憶領域に空き領域ができるまで)処理Aを実行することが不可能となる。したがって、結果的に、プロセッサ1にストール時間が生じる問題があった。

【0019】本発明は、上述した問題を解決するためになされたものであり、ストール時間を生じること無く、柔軟なスケジューリングを可能とすることによって高速に動作することが可能な並列処理方式を提供することを目的としている。

【0020】

【課題を解決するための手段】この目的を達成するために、本発明の請求項1に記載の並列処理方式は、処理手順の内容が同一あるいは異なる複数個の単体処理ユニットと、その複数個の単体処理ユニットの中から実行すべき単体処理ユニットを選択する単体処理ユニット選択手段とから構成される、複数個の複合処理ユニットと、前記複数個の前記複合処理ユニットを並列的に実行する並列処理制御手段とを備えることを特徴としている。

【0021】前記並列処理制御手段が複数個の複合処理ユニットを並列的に実行する際、個々の複合処理ユニットに於いて、単体処理ユニット選択手段は、複合処理ユニットを構成する複数個の単体処理ユニットの中から実行すべき単体処理ユニットを選択する。

【0022】したがって、各複合処理ユニットに於いて、前記単体処理ユニット選択手段により選択された単体処理ユニットが常に稼働していることになり、プロセ

4

ッサのストール時間が発生せず、柔軟なスケジューリングが可能な並列処理方式を実現することができる。

【0023】また、請求項2に記載の並列処理方式は、予め定義された優先順位である単体処理ユニット選択順序に従って、前記単体処理ユニット選択手段が前記複数個の単体処理ユニットの中から実行すべき単体処理ユニットを選択することを特徴としている。すなわち、単体処理ユニット選択手段は、前記単体処理ユニット選択順序に定義された優先順位に従って単体処理ユニットを選択する。したがって、処理の目的に合った並列処理のスケジューリングが可能となる。

【0024】請求項3に記載の並列処理方式に於いて、前記単体処理ユニット選択手段は、前記単体処理ユニット選択順序に従って選択された前記単体処理ユニットが実行可能であるか否かを判定する単体処理ユニット実行可能性判定手段と、実行可能な前記単体処理ユニットが選択されるまで単体処理ユニットの選択動作を行うように前記単体処理ユニット選択手段を制御する単体処理ユニット選択制御手段とを備えることを特徴としている。

【0025】すなわち、前記単体処理ユニット選択手段が、前記単体処理ユニット選択順序に定義された優先順位に従って単体処理ユニットを選択し、さらに、単体処理ユニット実行可能性判定手段が、選択された前記単体処理ユニットが実行可能であるか否かを判定し、単体処理ユニット選択制御手段が、前記単体処理ユニット実行可能性判定手段が実行可能であると判定するまで、前記単体処理ユニット選択順序に従って、順次前記単体処理ユニットの選択動作を繰り返す。

【0026】

【発明の実施の形態】以下、本発明を具現化した一つの実施の形態について、図面を参照しながら説明する。

【0027】図1は、本実施の形態の並列制御手段である並列処理装置100の回路構成の概略を示すブロック図である。尚、本並列処理装置100は、2個のCPUによって並列処理を実行する、いわゆる2CPUマルチプロセッサシステムとして構成されている。

【0028】前記並列処理装置100は、第1複合処理ユニット10を備えた第1CPU1と、第2複合処理ユニット20を備えた第2CPU2とから構成される。

【0029】前記第1複合処理ユニット10は、予め処理手順が定められた単体処理ユニットとしての第1処理手順11及び第2処理手順12と、単体処理ユニット選択手段としての第1処理選択手段13とから構成される。

【0030】前記第2複合処理ユニット20も同様に、予め処理手順が定められた単体処理ユニットとしての第1処理手順11及び第2処理手順12と、単体処理ユニット選択手段としての第2処理選択手段14とから構成される。

【0031】前記並列処理装置100が起動されると、

5

前記第1CPU1は、割り当てられた第1複合処理ユニット10の実行を開始する。これと並行して、第2CPU2は、割り当てられた第2複合処理ユニット20の実行を開始する。

【0032】前記第1CPU1に於いては、第1処理選択手段13によって第1処理手順11または第2処理手順12の何れかが選択され、選択された処理手順に沿って処理が実行される。第2CPU2に於いても、第2処理選択手段14によって第1処理手順11または第2処理手順12の何れかが選択され、選択された処理手順に沿って処理が実行される。

【0033】次に、図2及び図3を参照して、第1処理選択手段13及び第2処理選択手段14について詳細に説明する。

【0034】まず、図2を参照して、第1処理選択手段13について説明する。

【0035】その第1処理選択手段13には、第1処理手順11を優先して選択する単体処理ユニット選択順序が定められている。

【0036】前記第1処理選択手段13は、前記単体処理ユニット選択順序に従って、まず第1処理手順11を選択してその実行可能性を判定し、第1処理手順11が実行可能でない場合に第2処理手順12を選択してその実行可能性を判定する。したがって、前記第1処理選択手段13は、本発明の単体処理ユニット実行可能性判定手段をも兼ねる。

【0037】前記第1処理選択手段13は、最初に、第1処理手順11の実行可能性の判定を行う(S10)。

【0038】例えば、第2処理手順12が第1処理手順11の処理結果を利用して実行される場合、第2処理手順12の実行可能性の判定は、第1処理手順11の処理が終了しているか否かを判定することによって行われる。また、第1処理手順11が第2処理手順12の処理結果を利用して実行される場合、第1処理手順11の実行可能性の判定は、第2処理手順12の処理が終了しているか否かを判定することによって行われる。

【0039】尚、実行可能性の判定は、上述したような、処理結果の相互利用以外の要素、例えば、空きの記憶領域の存在等を考慮して行うこともできる。

【0040】具体例としては、例えば、第1処理手順11が前記処理B専用である場合、第1処理手順11の実行可能性は、処理しようとしている処理B(i)に利用されるべき前記処理A(i)の処理結果が何れかの記憶領域内にあるか否かにより判定される。また、第1処理手順11が前記処理A専用である場合、第1処理手順11の実行可能性は、処理しようとしている処理A(j)の処理結果を格納すべき空きの記憶領域が存在するか否かにより判定される。

【0041】上述のような実行可能性判定により、第1処理手順11の実行が可能であると判定された場合(S

6

10:Yes)、第1処理手順11が選択され(S12)、処理が終了する(S20)。

【0042】S10に於いて、第1処理手順11の実行が不可能であると判定された場合、(S10:No)、第2処理手順12の実行が可能か否かの判定が行われる(S14)。ここで、第2処理手順12の実行が可能であると判定された場合(S14:Yes)、第2処理手順12が選択されて(S16)、この処理が終了する(S20)。

【0043】S14に於いて、第2処理手順12の実行が不可能であると判定された場合には(S14:No)、第1CPU1で実行可能な処理手順が存在しないので、何も選択しないで(S18)、この処理を終了する(S20)。

【0044】第1処理選択手段13で何も選択されなかった場合は、第1CPU1は停止状態(ストール)となる。

【0045】同様に、図3を参照して、第2処理選択手段14について説明する。

【0046】その第2処理選択手段14には、第1処理手順11を優先して選択する単体処理ユニット選択順序が定められている。したがって、前記第2処理選択手段14は、前記単体処理ユニット選択順序に従って、まず第1処理手順11を選択してその実行可能性を判定し、第1処理手順11の実行が可能でない場合に第2処理手順12を選択してその実行可能性を判定する。したがって、前記第2処理選択手段14は、本発明の単体処理ユニット実行可能性判定手段をも兼ねる。

【0047】前記第2処理選択手段14は、最初に、第2処理手順12の実行可能か否かの判定を行う(S22)。

【0048】S22に於いて第2処理手順12の実行が可能であると判定された場合(S22:Yes)、第2処理手順12が選択され(S24)、処理が終了する(S32)。

【0049】S22に於いて第2処理手順12の実行が不可能であると判定された場合には(S22:No)、第1処理手順11の実行が可能か否かの判定が行われる(S26)。第1処理手順11の実行が可能であると判定された場合(S26:Yes)、第1処理手順11を選択して(S28)、この処理を終了する(S32)。

【0050】S26に於いて、第1処理手順11の実行が不可能であると判定された場合には(S26:No)、第2CPU2で実行可能な処理手順が存在しないので、何も選択しないで(S30)、この処理が終了する(S32)。

【0051】第2処理選択手段14で何も選択されなかった場合は、第2CPU2は停止状態(ストール)となる。

【0052】次に、上述した構成を有する、本実施の形

態の並列処理方式 100 の動作の様子を示す具体例について説明する。

【0053】図 4 は、図 1 の回路ブロック図と、図 2 及び図 3 のフローチャートを複合することにより、本並列処理装置 100 の動作の様子を説明するための図である。

【0054】ここで、本並列制御装置 100 を繰り返し起動し、処理 B の処理結果が 10 個得られた時点で処理を終了するものと設定する。

【0055】第 1 処理手順 11 は、図 5 のタイムチャートに示すように、10 t の処理時間を要する処理 A を実行し、第 2 処理手順 12 は、5 t の処理時間を要する処理 B を実行するものとする。また、前記第 2 処理手順 12 は、前記第 1 処理手順 11 の処理結果を使用して処理を実行するものとし、前記第 1 処理手順 11 の処理結果が得られたことを条件として実行可能となるものとする。

【0056】すなわち、前記第 2 処理手順 12 に於いて処理される i 番目の処理 B である処理 B (i) は、前記第 1 処理手順 11 による処理 A (i) の結果を利用するものである。したがって、処理 B (i) は処理 A (i+1) の前に行くことはできるが処理 A (i) の前に行くことはできない。また、処理 A (i) は、記憶領域に格納できる限り、いずれの処理 B にも先だてて行うことができる。

【0057】また、単体処理ユニット選択順序は、第 1 複合処理ユニット 10 に於いては第 1 処理手順 11 が優先、第 2 複合処理ユニット 20 に於いては第 2 処理手順 12 が優先とする。さらに、処理 A の処理結果は、3 個まで記憶領域に格納できるとする。

【0058】はじめに、第 1 CPU 1 は第 1 処理手順 11 を選択して処理 A (1) を実行する。一方、第 2 CPU 2 では、処理 B (1) を実行することが不可能なため、第 2 処理選択手段 14 によって第 2 処理手順 12 が選択され、処理 A (2) を実行する。前記処理 A (1) 及び A (2) が終了する時点は、本並列制御装置 100 の作動開始より 10 t 経過後である。

【0059】次に、第 1 CPU 1 は処理 A 3 を実行する。これと並行して、第 2 CPU 2 では、処理 B (1) から B (2) までが実行可能となったため、処理 B を実行し、処理 B (1) 及び B (2) の処理結果を得る。以降、同様の手順で処理を続行する。

【0060】以上の動作を行った結果のタイムチャートが図 5 である。全体の処理時間は 75 t であり、第 1 CPU 1 を処理 A、第 2 CPU 2 を処理 B に固定して並列処理した場合 (105 t) よりも約 30% の処理時間短縮となる。しかも、柔軟なスケジューリングにより、両 CPU に於けるストール時間の発生はない。

【0061】さらに、上述の実施の形態の変形として、単体処理ユニットを 3 個備えた CPU を 2 個用いた場合

の動作について説明する。

【0062】基本的な全体構成は図 4 と同じであるが、第 1 複合処理ユニット 10 及び第 2 複合処理ユニット 20 を構成する単体処理ユニットの数が各々 3 個になる部分が異なる。

【0063】上述の具体例と同様に、並列処理装置 100 を繰り返し起動し、処理 C の処理結果が 5 個得られた時点で処理を終了するものと設定する。

【0064】また、処理 A は 10 t、処理 B は 5 t、処理 C は 15 t の処理時間を要し、処理 B は処理 A の処理結果を使用し、処理 C は処理 B の処理結果を使用するものと仮定する。

【0065】さらに、単体処理ユニット選択順序は、第 1 複合処理ユニット 10 に於いては処理 A、処理 B、処理 C の順序となり、第 2 複合処理ユニット 20 に於いては処理 C、処理 B、処理 A の順序となるように設定するものとし、処理 A、処理 B の処理結果は、各 3 個まで記憶領域に格納できるものとする。

【0066】以上の条件の下で動作を行った結果のタイムチャートを図 6 に示す。全体の処理時間は 80 t となる。処理後半 (45 t 以降) では、処理 A を全て終了した第 1 CPU 1 が、優先順位に従って処理対象となる処理手順を選択し、第 2 CPU 2 と共に、残りの実行可能な処理を行っていることが確認できる。このように、本実施の形態の並列処理方式によれば、各 CPU で柔軟に処理手順の割り当てを行うことが可能であるため、全体の処理時間が短縮できる。

【0067】次に、本実施の形態の具体的な適用例として、ビットマップイメージの描画処理に適用した場合の構成と動作について図面を参照して説明する。

【0068】図 7 は、ビットマップイメージの描画処理の概略手順を示す図である。

【0069】ビットマップイメージデータ 40 が入力されると、画素データ取得処理 (PXL) 50 によって 1 画素のデータを取得する。続いて、色変換処理 (CLR) 51 によって所望の色変換が施された後、塗りつぶし領域決定処理 (RGN) 52 によって色変換された画素データを塗りつぶす領域が決定され、ハーフトーン描画処理 (DRW) 53 によって、色変換された画素データをハーフトーン処理によって 2 値化し、先に決定された塗りつぶし領域に描画し、プリントデバイス 55 に出力する。

【0070】以上の処理を、入力されたビットマップイメージデータ 40 の全ての画素データについて繰り返すことによって、ビットマップイメージの描画処理が実行される。尚、ハーフトーン描画処理 53 の部分は、出力先のプリントデバイス 55 に応じてフルカラー描画処理や誤差拡散描画処理等に入れ替えが可能である。

【0071】図 8 は、図 7 に示すビットマップイメージ描画処理を、各処理過程での処理結果を格納する記憶領

9

域（バッファ）を使用することにより並列処理可能に構成した処理手順を示す図である。

【0072】画素データ取得処理50によって取得された画素データは、画素データバッファ60に格納される。色変換処理51は、画素データバッファ60から画素データを読み出し、色変換処理を行って結果を色変換バッファ61に格納する。塗りつぶし領域決定処理52は、画素データの取得とは独立して実行可能である。塗りつぶし領域決定処理52で決定された領域データは、領域データバッファ62に格納される。ハーフトーン描画処理53は、色変換バッファ61から色変換された画素データを読み出してハーフトーン処理によって2値化し、領域データバッファ62から塗りつぶし領域データを読み出して、塗りつぶし領域に描画し、プリントデバイス55に出力する。

【0073】図9は、図8の構成を上述の実施の形態の並列処理方式に適用した場合の動作の詳細を説明するための図であり、図4と同様に、回路ブロック図とフローチャートとを組み合わせたものである。

【0074】第1CPU1に割り当てられた第1複合処理ユニット80は、画素データ取得処理50、色変換処理51、塗りつぶし領域決定処理52、ハーフトーン描画処理53の各単体処理ユニット、及び第1処理選択手段70より構成される。

【0075】前記第1処理選択手段70に於ける単体処理ユニット選択順序は、DRW、CLR、PXL、RGNの順に設定されている。

【0076】同様に、第2CPU2に割り当てられた第2複合処理ユニット81は、画素データ取得処理50、色変換処理51、塗りつぶし領域決定処理52、ハーフトーン描画処理53の各単体処理ユニット、及び第2処理選択手段71より構成される。

【0077】前記第2処理選択手段71に於ける単体処理ユニット選択順序は、DRW、RGN、CLR、PXLの順に設定されている。

【0078】したがって、第1CPU1では主として画素データの取得、色変換の処理を行い、第2CPU2では主として塗りつぶし領域決定処理とハーフトーン描画を行うような優先順位が設定されている。

【0079】ここで、DRW、CLR、PXL、RGNの各処理時間が同じであると仮定した場合の実行結果のタイムチャートを図10に示す。

【0080】最初、第2CPU2に於いては、RGNが、領域データバッファ62の記憶領域が無くなるまで連続的に実行される。これと並行して、第1CPU1ではPXL、CLR、DRWが順に繰り返し実行される。領域データバッファ62の記憶領域が無くなった時点で、第1CPU1、第2CPU2は、記憶領域に格納されたデータを消費すべく、PXL、CLR、DRWの処理を実行する。

10

【0081】すなわち、先に処理できるものは、記憶領域の許容範囲で先に処理を済ませておき、ある時点で足りないデータが生じた場合は、全てのCPUパワーを投入して不足データの作成に対応する。これらのサイクルを繰り返すことで、CPUを休みなく動作させることが可能となり、高速な並列処理によるビットマップイメージ描画処理が実現されることになる。

【0082】尚、本発明は以上に説明した実施の形態に限定されるものではなく、本発明の要旨を変更しない範囲内に於いて、種々の変形が可能である。

【0083】例えば、前記実施の形態に於いては、各複合処理ユニットを構成する単体処理ユニットは2つ若しくは3つずつであり、その処理内容も各複合処理ユニットで共通であったが、複合処理ユニットを構成する単体処理ユニットの数は2個以上であれば特に制限がなく、また、各複合処理ユニットに備えられる単体処理ユニットの処理内容が互いに異なってもよい。

【0084】また、単体処理ユニット選択順序は、前記実施の形態の如く処理順序を予め定義したものに限定されず、処理順序を動的に決定する方法を予め定義したものでよい。後者のものによれば、単体処理ユニットの処理負荷や処理内容が何らかの原因で変動することがあっても、単体処理ユニット選択順序に定められた優先順位を柔軟に変更することが可能となる。

【0085】

【発明の効果】以上に説明したことから明かなように、本発明の請求項1に記載の並列処理方式によれば、並列処理制御手段が複数個の複合処理ユニットを並列的に実行する際、個々の複合処理ユニットに於いて、単体処理ユニット選択手段が、複合処理ユニットを構成する複数個の単体処理ユニットの中から実行すべき単体処理ユニットを選択する。

【0086】したがって、各複合処理ユニットに於いて、前記単体処理ユニット選択手段により選択された単体処理ユニットが常に稼動していることになり、プロセッサのストール時間が発生せず、柔軟なスケジューリングが可能な並列処理方式を実現することができる。

【0087】また、請求項2に記載の並列処理方式によれば、予め定義された優先順位である単体処理ユニット選択順序に従って、前記単体処理ユニット選択手段が前記複数の単体処理ユニットの中から実行すべき単体処理ユニットを選択する。すなわち、単体処理ユニット選択手段は、前記単体処理ユニット選択順序に定義された優先順位に従って単体処理ユニットを選択する。したがって、処理の目的に合った並列処理のスケジューリングが可能となる。

【0088】さらに、請求項3に記載の並列処理方式によれば、前の処理結果を利用した後の処理を実行する場合、後の処理が実行可能ならばその処理を実行し、前の処理が終了していないために後の処理が不可能な場合

11

は、前の処理を選択して実行することが可能となる。すなわち、前の処理が遅れた場合には、処理の遅い前処理を複数のプロセッサで処理し、プロセッサのストールを防ぐことができる。

【0089】したがって、ストール時間を生じること無く、柔軟なスケジューリングを可能とすることによって高速に動作することが可能な並列処理方式を提供することができる。

【図面の簡単な説明】

【図1】本発明を具現化した一つの実施の形態の並列処理方式の回路ブロック図である。

【図2】前記並列処理方式に備えられる第1処理選択手段に於ける処理を説明するためのフローチャートである。

【図3】前記並列処理方式に備えられる第2処理選択手段に於ける処理を説明するためのフローチャートである。

【図4】前記並列処理装置の動作を説明するための図である。

【図5】前記並列処理装置による動作結果を示すタイムチャートである。

12

*【図6】前記並列処理装置を変形した並列処理方式の動作結果を示すタイムチャートである。

【図7】ビットマップイメージ描画処理の手順の概略を説明するための図である。

【図8】前記ビットマップイメージ描画処理の手順の詳細を説明するための図である。

【図9】前記ビットマップイメージ描画処理に前記実施の形態の並列処理方式を応用した場合の回路ブロック図である。

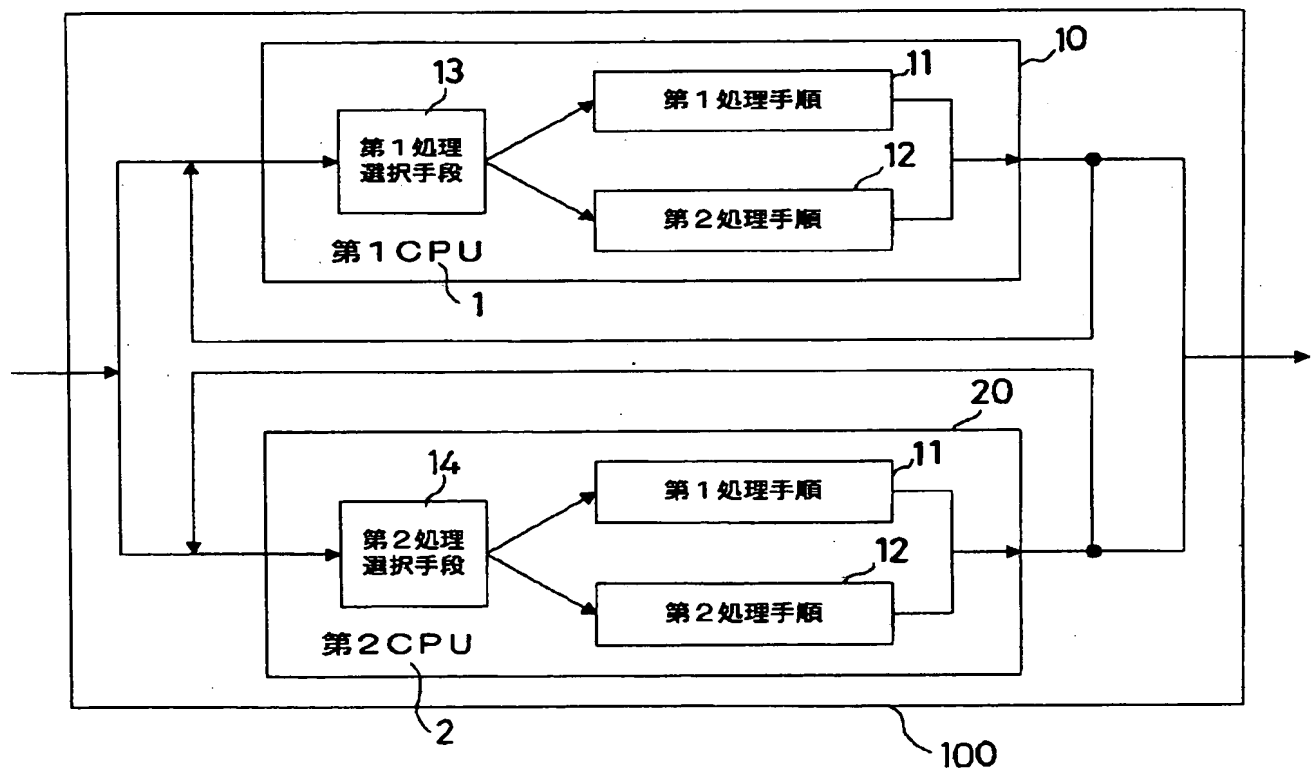
【図10】前記並列処理方式によるビットマップイメージ描画処理の動作結果を示すタイムチャートである。

【図11】従来の並列処理方式の動作結果を示すタイムチャートである。

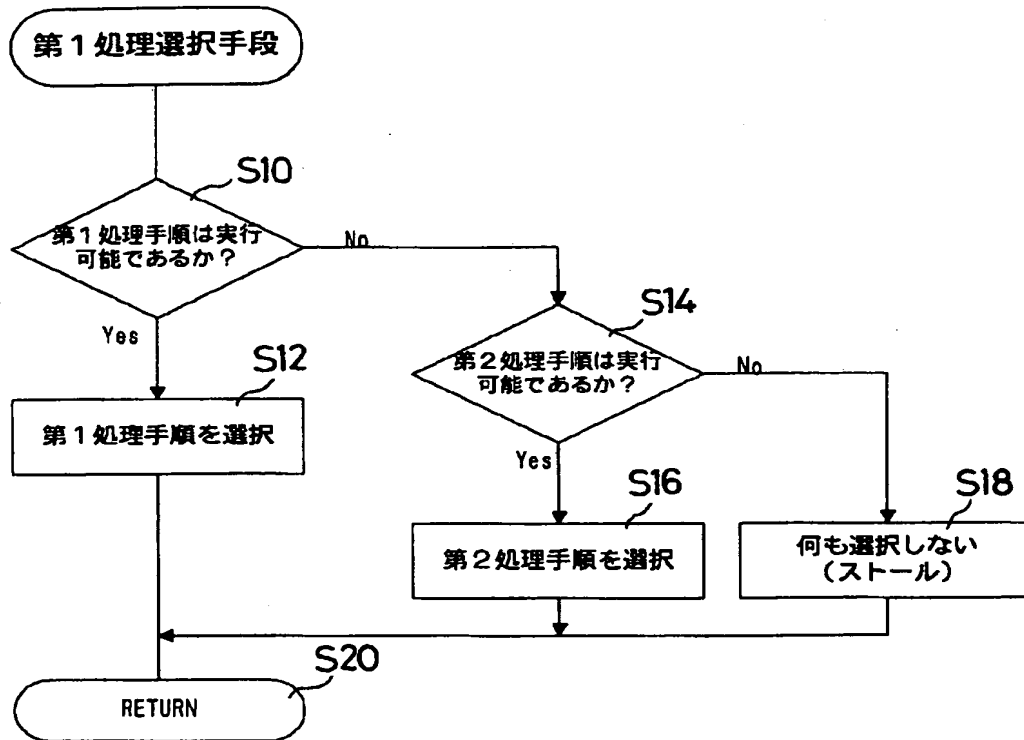
【符号の説明】

- | | |
|-----|------------|
| 11 | 第1処理手順 |
| 12 | 第2処理手順 |
| 10 | 第1複合処理ユニット |
| 20 | 第2複合処理ユニット |
| 13 | 第1処理選択手段 |
| 14 | 第2処理選択手段 |
| 100 | 並列処理装置 |

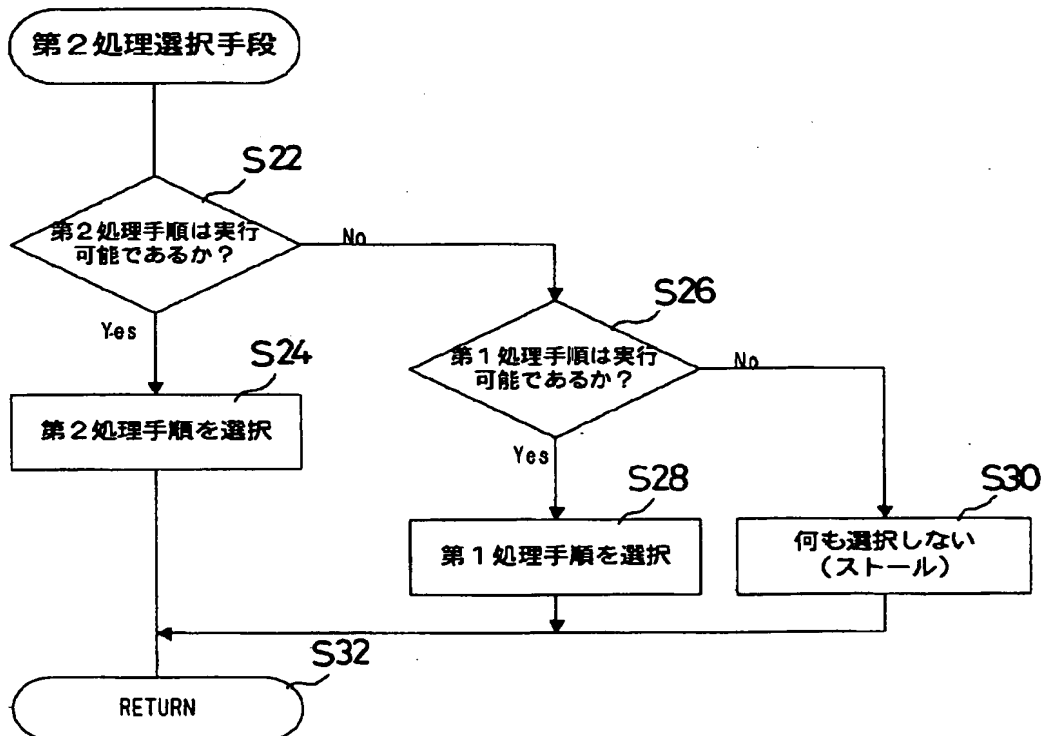
【図1】



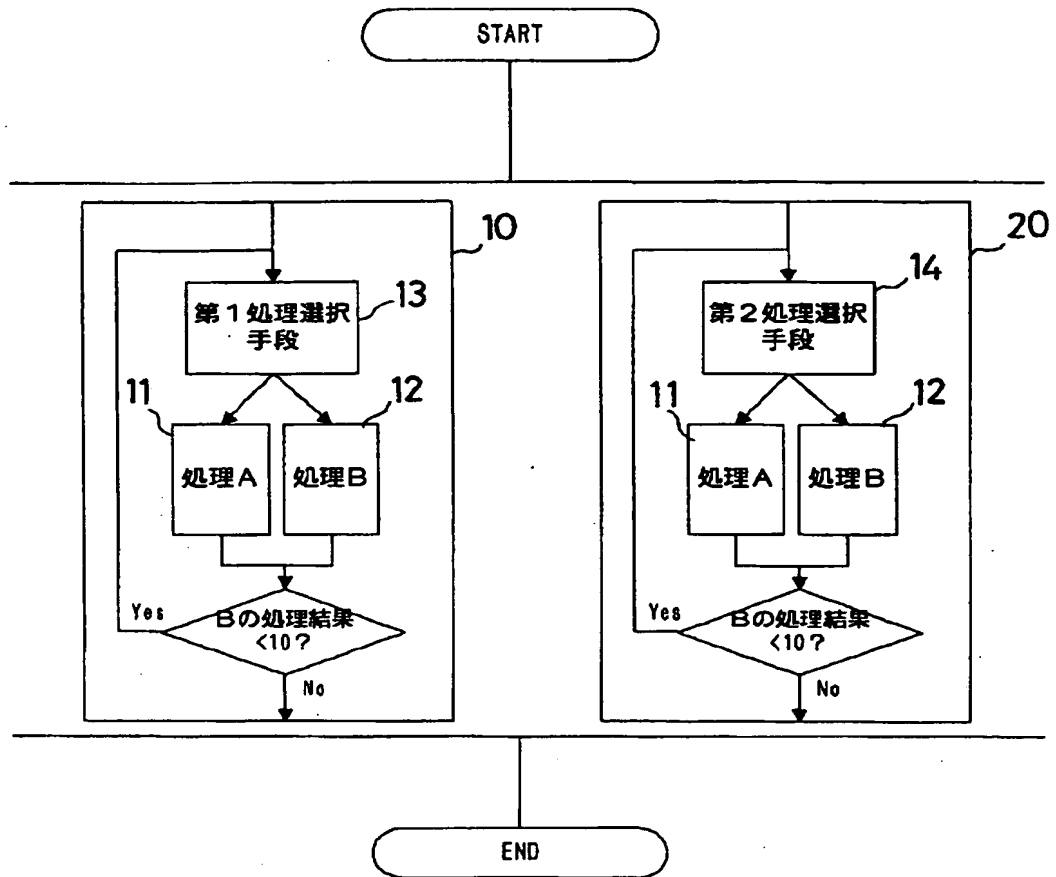
【図2】



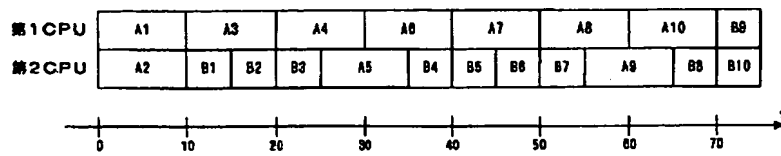
【図3】



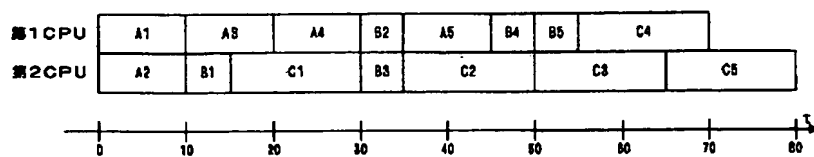
【図4】



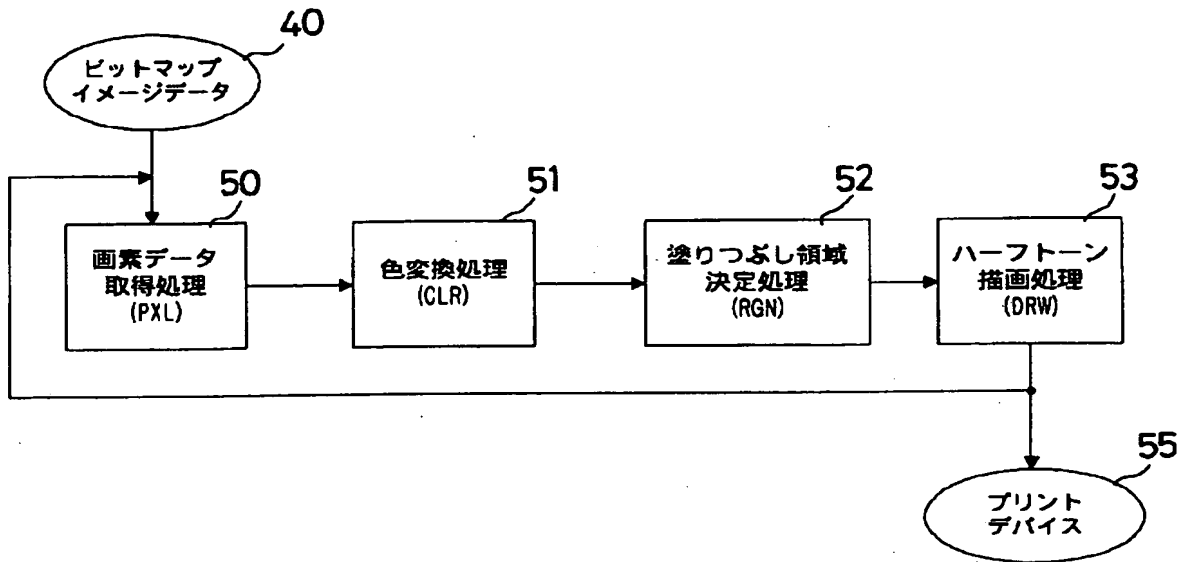
【図5】



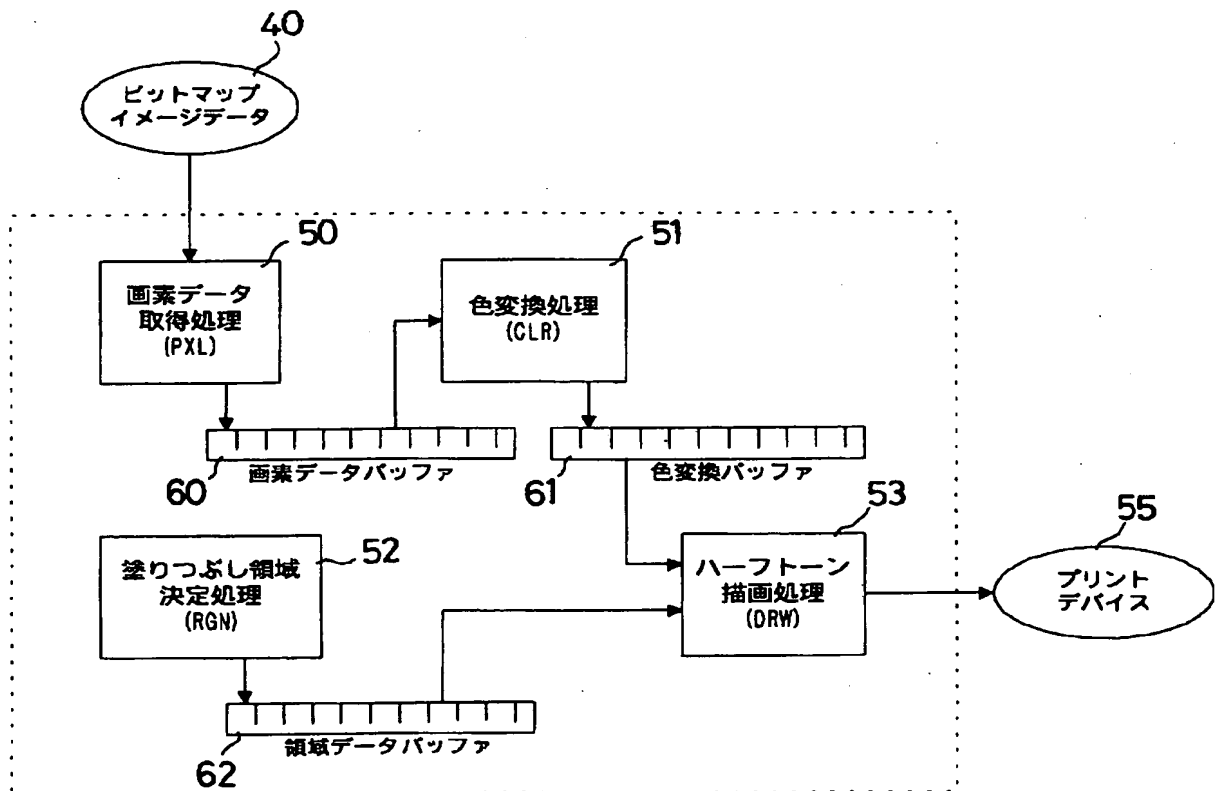
【図6】



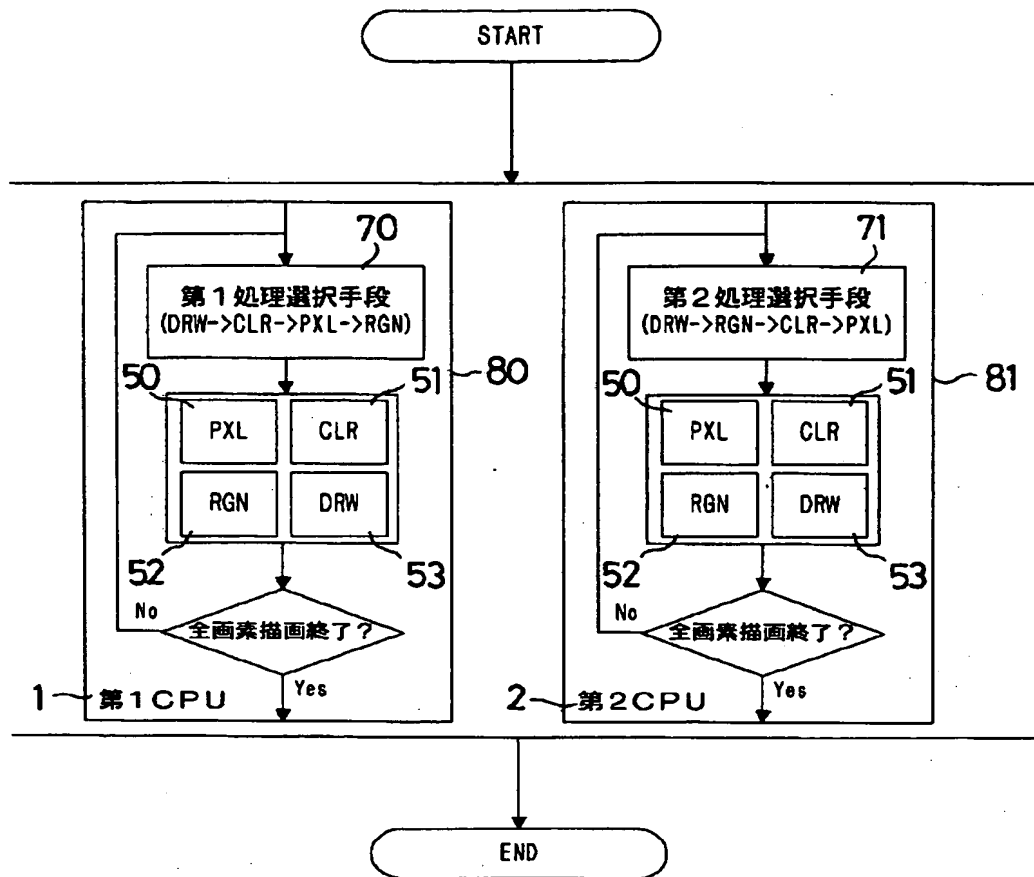
【図 7】



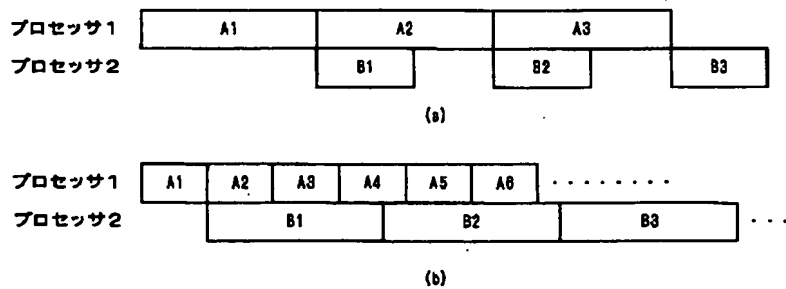
【図 8】



【図9】



【図11】



【図10】

